

Proceedings of The Canadian Society for Mechanical Engineering International Congress 2018
CSME International Congress 2018
May 27-30, 2018, Toronto, On, Canada

SENSOR FILTERING AND POSITION TRACKING OF A SIMULATED PLANAR BIPEDAL ROBOT

Stefano Rossi* and S. Andrew Gadsden
College of Engineering and Physical Sciences,
University of Guelph,
Guelph, Ontario, Canada
*srossi@uoguelph.ca

Abstract—The development of a humanoid robot is a very prevalent area of research today. Legged robots have many advantages over wheeled robots on rough or uneven terrains, and are more suitable for an everyday household setting, however they possess many design and control challenges. The spring loaded inverted pendulum (SLIP) is frequently used as a fundamental model to analyze bipedal locomotion. In general, it consists of a stance phase and a flight phase, employing different strategies during these phases to control speed and orientation. Due to the underactuation and hybrid dynamics of bipedal robots, estimating the state of the robot and its appendages can be challenging. In this paper, various Kalman estimation techniques are used to predict the state of a simulated planar SLIP model. The state estimations utilize only simulated sensor data, as if the simulated model was a physical one.

Keywords—*Kalman filter; bipedal robot; legged robot; robotics; simulation; inverted pendulum; SLIP; hybrid dynamics; advanced estimation; sensor filtering.*

I. INTRODUCTION

The simulated model consists of a planar biped robot designed and simulated in Virtual Robotic Experimentation Platform (VREP), using Open Dynamics Engine (ODE) for the physics. In flight phase, when there is no ground contact, the dynamics of the system is simply ballistic flow and has well known and analytically solved dynamics. However, due to gravity, the stance phase dynamics is much more complicated and includes coupled equations and non integratable terms. Therefore, an accurate estimation of the state based on sensor measurements becomes much more valuable than closed form analytical motion planning. The main controller design goals of the robot are:

1. To achieve dynamically stable running.
2. To be able to accelerate and decelerate to achieve a desired running speed.
3. To control its angular momentum to maintain a desired body attitude during running.

These design goals are achieved through the main control loop, however the controller relies on accurate measurements of the body attitude, leg positions and angular and linear velocity

of the robot. The Kalman filter (KF) is applied to estimate positions and velocities of these components [1]. The KF is an optimal state estimation strategy, and is widely used in the field of control and estimation theory [2, 3].

A. SLIP Model

To study running in its simplest form, a planar running machine was built at MIT [4]. This robot only had one leg, but the principle is identical to a biped. The leg behaved as a passive spring and was telescopic. It used pneumatics to simulate a spring and could also apply thrust force. During one cycle there was a stance and flight phase. During the stance phase, the leg supports the body and remains in a fixed location on the ground. In this phase, the robot tips like an inverted pendulum and there is no chance to move the foot placement to control position. In order to change the foot position, the robot jumps to flight phase where the leg is unloaded and free to move. Marc Raibert developed a simple control strategy for simple legged robots which decomposes into the parts, the hopping height, the forward speed and the body attitude [5].

B. Previous Research

Many methods of state estimation for planar legged locomotion exist currently [6]. Stance mapping is a commonly used approach in state estimation of linkages. It has been proven that stance dynamics are nonlinear, coupled and contain non-integratable terms [7]. Many assumptions and approximations need to be made to derive a stance map. Common assumptions include ignoring gravity, no inputs, constant speed, small spring compressions and small sweep angles [8]. Other methods use a mean value theorem derived from interactive performance testing [9]. Kalman filtering has also extensively been applied to the linkages of bipedal walking models [10], however a walking model lacks the complex hybrid dynamics of running, and again many assumptions are made to reduce the model, such as constant height CoG [6]. Applying these methods to real life models is challenging and impractical, with the introduction of system and measurement noise, uncertain impact dynamics and imperfect limb coordination and model design. An alternative to a closed form stance map is real-time adaptive state estimation from sensor data, which is implemented in this project.

C. Kalman Filter

In order for the Raibert controller to function, the orientation, linear and angular velocity of the main body needs to be known, as well as the positions of the legs relative to the body. Two Kalman filters are applied to the entire model to determine the state relative to the robots' constituents. The first is used to determine the attitude of the body of the model. It utilizes a gyroscope and the control torque applied during stance. The second is a sensor fusion Kalman filter utilizing an accelerometer and gyroscope on the pivot points of the leg.

II. SIMULATED MODEL

The model created in VREP originally based on Raibert planar biped [10]. The model consists a rectangular main body, with two actuated hip joints, which connect to telescopic legs (Fig. 1). The legs act as a passive spring/dampers during compression, and are capable of applying a thrust force. The robot is attached to a spherical joint at the hip by a 5m massless boom (Fig. 1). This eliminates 3DOF from the model, its yaw, roll and lateral movement. The feet consist of spheres and have perfect friction with the ground ($\mu = 1$).

TABLE I. SYMBOLS AND DESCRIPTIONS

Symbol	Description
θ	Leg angle relative to body vertical
φ	Body angle with respect to horizontal
φ_d	Desired body angle
r	Leg length at equilibrium
z	Body height from contact surface
x	Body position on track
\dot{x}	Body velocity
\dot{x}_d	Desired body velocity
x_r	Foot distance from body CoG
M_B	Mass of body
M_L	Mass of leg
I_B	Body mass moment of inertia
g	Gravity
T_{st}	Time of Stance
K_L	Spring constant of leg
τ	Control torque of active hip

The simulation used Open Dynamics Engine as its physics engine due to its accuracy modelling spring damper systems. The main control loop is implemented directly in VREP via a child script. The running motion of the robot can be described in 5 phases: flight, touchdown, compression, thrust and takeoff. The main working principle behind the speed control is foot placement. Because the leg acts as a spring-damper, the time of the stance phase (compression and thrust) can be approximated by:

$$T_{st} = \frac{\pi}{\omega} = \pi \sqrt{\frac{M_B + M_L}{K_L}} \quad (1)$$

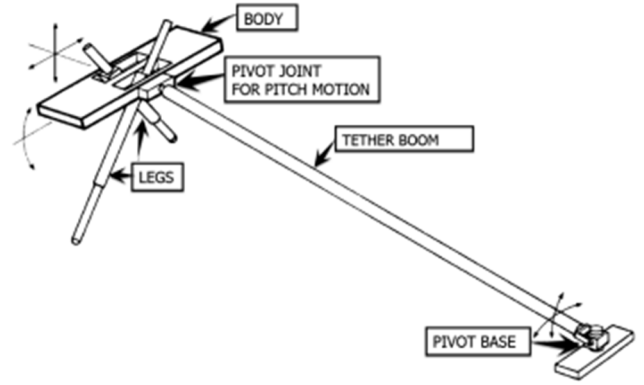


Figure 1. Biped model assembly.

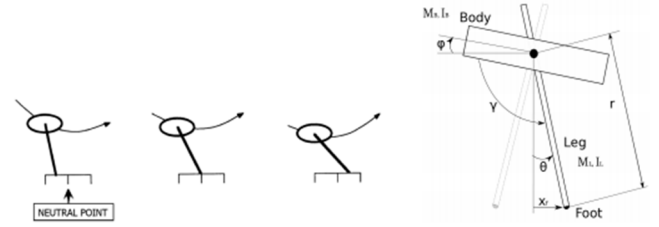


Figure 2. Foot placement effects and free body diagram.

Foot placement has a direct effect on the resultant velocity at takeoff. If the foot is placed directly at the halfway point throughout the stance (neutral point), the stance phase is symmetric and the takeoff velocity is the same as the touchdown.

$$x_{f0} = \frac{\dot{x}T_{st}}{2} \quad (2)$$

Any deviation from the neutral point results in a non-zero horizontal acceleration. Placing the foot before the neutral point results in acceleration, as more of the vertical velocity is converted to horizontal, and vice versa (Fig. 2). To accelerate to a desired speed, proportional control is used.

Therefore, the algorithm for foot placement and the corresponding hip angle is:

$$x_f = \frac{\dot{x}T_{st}}{2} + k_x(\dot{x} - \dot{x}_d) \quad (3)$$

$$\gamma_d = \varphi - \sin^{-1}\left(\frac{\dot{x}T_{st}}{2} + \frac{k_x(\dot{x} - \dot{x}_d)}{r}\right) \quad (4)$$

Body attitude is maintained by applying a torque about the hip during the stance phase. Since angular momentum is conserved during flight, the friction between the foot and the ground provides an opportunity to correct the angular momentum of the entire system. To servo the body to a desired attitude, this control torque is applied:

$$\tau = -k_p(\varphi - \varphi_d) - k_v(\dot{\varphi}) \quad (5)$$

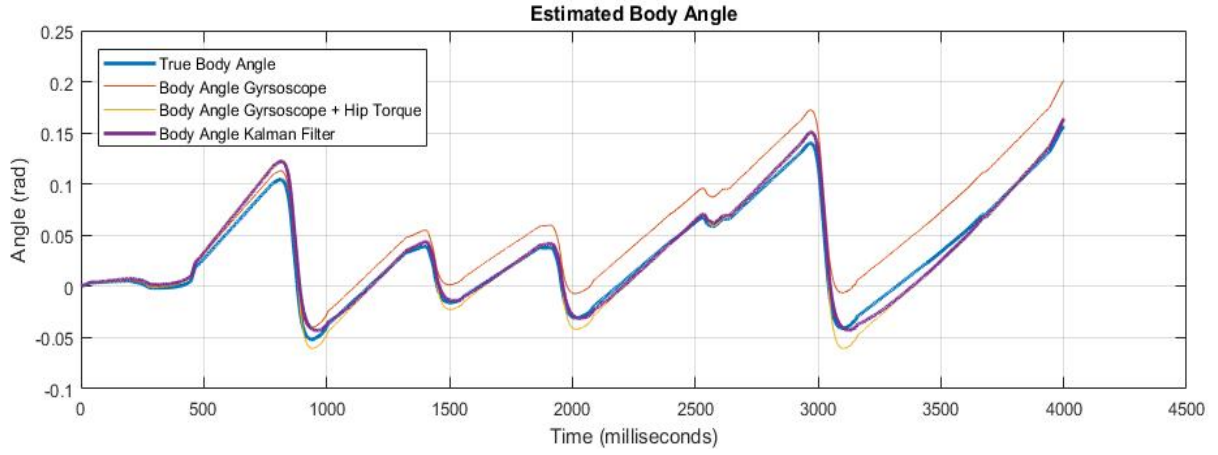


Figure 3. True and estimated body angle.

where k_p and k_v are constants, 80 and 20 respectively. During the stance and flight phases, the idle leg mirrors the active leg to cancel out any angular momentum.

III. BODY STATE ESTIMATION

The body angle of the robot is determined through a gyroscope and the applied control torque during stance phase. The gyroscope was simulated using a reference point where the Euler angles were read, and simulation noise was added. The control torque is an input based on the current body angle and angular velocity. These values were streamed from VREP in real time to MATLAB using a remote API. They were then filtered and streamed back into VREP to calculate the control inputs for the next time step. The true angle and angular velocity of the body were also streamed to MATLAB for comparison.

TABLE II. SIMULATION PARAMETERS

Parameter	Value	Description
t	0.001	Simulation time step, in sec
I	4.708e-02	Body mass moment of inertia
ϕ_d	0	Desired body angle, in rad
r	0.5575	Leg length uncompressed
\dot{x}_d	2	Desired body velocity, in m/s
M_B	9.246	Mass of body, in kg
M_L	0.478	Mass of leg, in kg
T_{st}	0.178	Time of Stance, in sec
K_L	2700	Spring constant of leg, in N/m

A. Estimation Methods

Three methods were used to estimate the angle of the body. The angle was estimated just from the gyroscope measurement of the angular velocity.

$$\phi_i = \phi_{i-1} + t\dot{\phi}_i \quad (6)$$

The body angle was also calculated using the input torque applied during the stance phase to correct the body angle.

$$\phi_i = \phi_{i-1} + t\dot{\phi}_i + \frac{\tau}{2I}t^2 \quad (7)$$

The body angle was also estimated using a Kalman filter.

$$\begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} \frac{t^2}{2I} \\ t/I \end{bmatrix} \begin{bmatrix} \tau \end{bmatrix} \quad (8)$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (9)$$

Measurement and system noise were then added to the system to simulate real world conditions. These values were assumed to be Gaussian and white. These values were approximated, and it was assumed most of the noise would stem from the sensors, so the system noise was several orders of magnitude less than the measurement noise.

B. Results

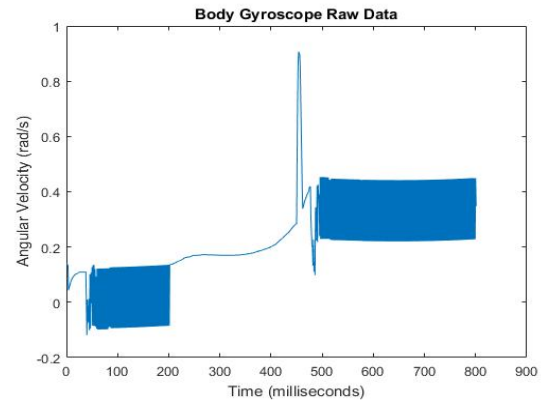


Figure 4. Body gyro data without Gaussian noise.

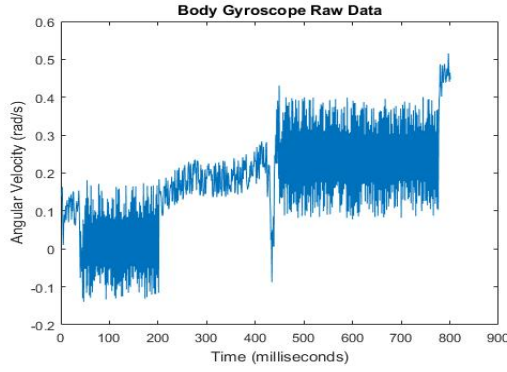


Figure 5. Body gyro data with Gaussian noise.

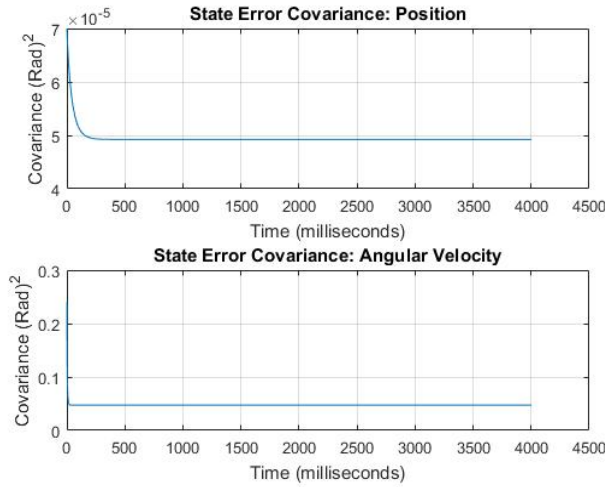


Figure 6. Position and angular velocity state error covariance.

TABLE III. ROOT MEAN SQUARE ERROR RESULTS

	Gyroscope Angle Estimation (rad ²)	Gyro + Torque Angle Estimation (rad ²)	Kalman Filter Angle Estimation (rad ²)
RMSE	0.0393433	0.0184116	0.0071413

The KF performed the best with an RMSE approximately five times lower than that of the gyroscope estimation alone, and 2.5 times lower than that of the gyroscope and the torque estimation. As per the position estimation results, the gyroscope estimation drifts over time, while the combined estimation overshoots the extreme tilt angles. The KF accurately accounts for these deviations and best represents the true angle of the body. Although the Kalman filter is able to filter out most of the noise in the angular velocity estimation, it does seem to have a bias. This may be due to the Kalman filter counteracting the bias of the gyroscope, as the state error covariance of both the position and the velocity converge quickly over time. The estimated state of the body was streamed back into the VREP simulation to predict the next control outputs, and the robot performed just as well as if reading the true states, running at its desired speed of 2m/s with a deviation of 0.2m/s.

IV. LEG STATE ESTIMATION

The state estimation of the leg consists of a sensor fusion between an accelerometer and gyroscope located at the hip joint of the model. The virtual accelerometer consists of a reference mass attached to a force sensor. These values were streamed from VREP in real time to MATLAB using a remote API. They were then filtered and streamed back into VREP to calculate the control inputs for the next time step. The true angle and angular velocity of the leg were also streamed to MATLAB for comparison.

A. Estimation Methods

Due to the impact forces of touchdown and the rising and falling motion of the model, the calculated accelerometer angle of the leg is very noisy and very biased during the stance phase (Fig. 7). The signal is processed through two corrective algorithms before it is filtered to provide better estimations. The first one uses accelerometer data from the body rather than the leg. Because the body of the biped remains relatively level (± 5 deg) compared to the sweep range of the leg (± 40 deg), the calculated accelerometer angle of the body is subtracted from that of the leg to correct for the rising and falling motion of the model. The data is then corrected for the large spikes occurring during touchdown (Fig. 8). The angle of the leg was estimated using two filters, a complimentary and a Kalman. The complementary filter combined the calculated angle from the gyroscope sensor and the accelerometer sensor.

$$\theta_{comp} = \alpha(\theta_{gyro}) + (1 - \alpha)(\theta_{accel}), \alpha = 0.9 \quad (10)$$

The state of the leg was also estimated using a Kalman filter. Measurement and system noise were then added to the system to simulate real world conditions. These values were assumed to be Gaussian and white. These values were approximated, and it was assumed most of the noise would stem from the sensors, so the system noise was several orders of magnitude less than the measurement noise.

$$\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \varphi \\ \dot{\varphi} \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (11)$$

$$Q = \begin{bmatrix} \frac{t^3}{3} & \frac{t^2}{2} \\ \frac{t^2}{2} & t \end{bmatrix}, R = \begin{bmatrix} 10^{-4} & 0 \\ 0 & 10^{-1} \end{bmatrix} \quad (12)$$

B. Results

As seen in the results (Fig. 10 & 11), the KF outperformed the complementary filter, with an RMSE less than half that of the complementary (Table IV). The complementary performed fairly well, however after a certain time the gyroscopic drift had too large of an effect on the estimation. The Kalman filter was able to filter out much of the noise from the accelerometer reading (Fig. 10). While the position tracking estimation was smooth, very little of the velocity estimation noise was filtered out. This is primarily due to the low measurement noise covariance applied to the gyroscope data, and a large noise covariance applied to the accelerometer. The estimated state of the active leg was streamed back into the VREP simulation to predict the next control outputs in real time. The robot performed just as well as if reading the true states, running at its desired speed of 2m/s with a deviation of 0.2m/s.

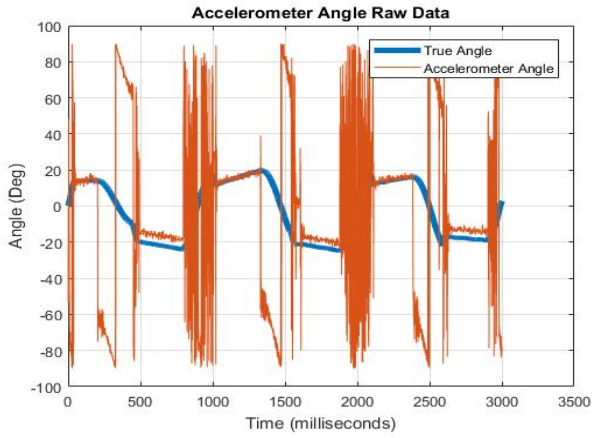


Figure 7. Raw accelerometer data from leg.

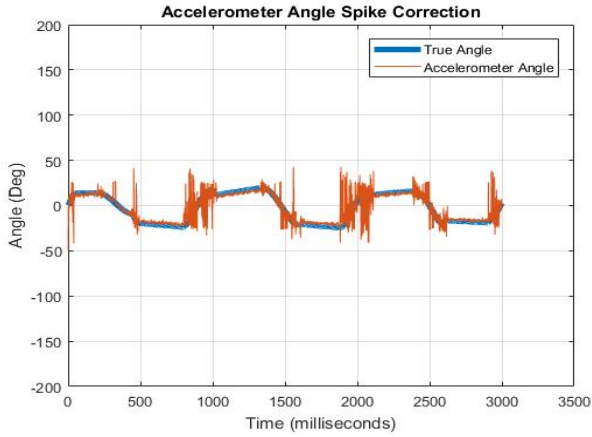


Figure 8. Filtered accelerometer data from leg.

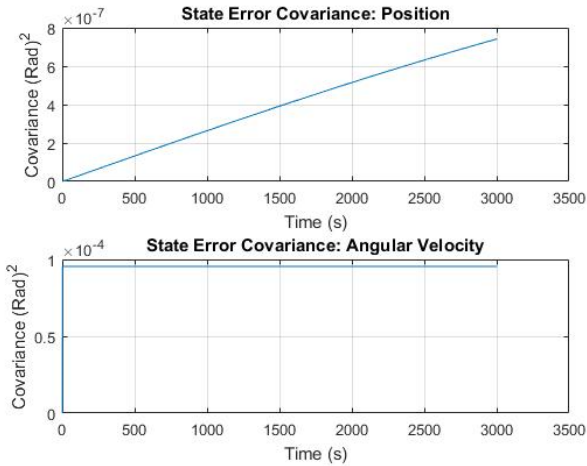


Figure 9. State error covariance of leg estimation.

TABLE IV. ROOT MEAN SQUARE ERROR RESULTS

	Complementary Filter Angle Estimation (rad ²)	Kalman Filter Angle Estimation (rad ²)
RMSE	0.0562632	0.0276804

V. CONCLUSIONS

Experimentation showed that the implementation of a Kalman filter is very effective at estimating of the states of the model. The Kalman filter worked best when the system was linear and known. The system matrix has the largest effect on the performance of the Kalman filter. Despite the challenges of predicting the overall state of the model, it has been demonstrated that it is possible to use Kalman filtering techniques to estimate the state of this model just using sensor data, allowing for a physical implementation.

VI. FUTURE WORK

While the Kalman estimations for the states of the body and the legs were very successful, the state estimation of the entire model did not yield desirable results. Linearized state equations for the stance phase would be very useful in improving the filters, and allowing the implementation of an EKF and IMM-EKF strategy. Although the simulated model is easier to work with, a physical model would be the next step in experimentally tuning these filters. Ideally, one large filter would predict the changing center of gravity and hybrid dynamics would be state of the entire system, however the equations integrating highly complex. Finally, this system lends itself to a neural net configuration. Applying a Kalman filter to adaptively train the neural net may yield a very successful control strategy.

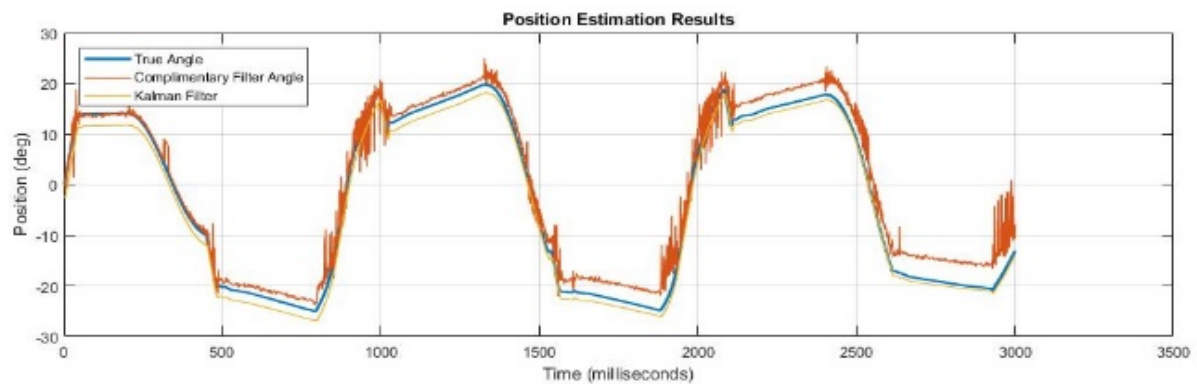


Figure 10. Position estimation results for leg angle.

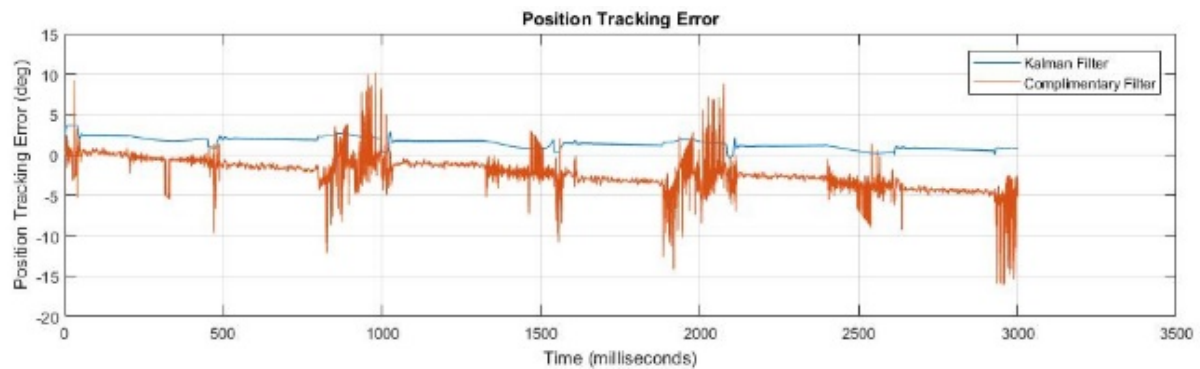


Figure 11. Position tracking error for leg angle estimation.

REFERENCES

- [1] S. A. Gadsden and A. S. Lee, "Advances of the Smooth Variable Structure Filter: Square-Root and Two-Pass Formulations," *Journal of Applied Remote Sensing*, vol. 11, no. 1, pp. 1-19, 2017.
- [2] H. H. Afshari, S. A. Gadsden and S. R. Habibi, "Gaussian Filters for Parameter and State Estimation: A General Review and Recent Trends," *Signal Processing*, vol. 135, pp. 218-238, 2017.
- [3] S. A. Gadsden, Y. Song and S. R. Habibi, "Novel Model-Based Estimators for the Purposes of Fault Detection and Diagnosis," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, 2013.
- [4] M. H. Raibert, "Hopping in Legged Systems-Modeling and," *IEEE Transactions On Systems, Man, and Cybernetics*, vol. 14, p. 3, 1984.
- [5] M. H. Raibert, *Legged Robots That Balance*, Cambridge, Mass.: MIT Press, 1986.
- [6] Xinjilefu, "State Estimation for Humanoid Robots," Carnegie Mellon University Robotics Institute, Pittsburgh, Pennsylvania, 2015.
- [7] A. Omur, "Approximate Stance Map For Nonsymmetric Motions and Variable Stiffness," Electrical & Electronics Engineering Department, Bilkent University, 2008.
- [8] H. Geyera, "Spring-Mass Running: Simple Approximate Solution and Application," *Journal of Theoretical Biology*, vol. 232, p. 315-328, 2005.
- [9] W. J. Schwind, "Spring loaded inverted pendulum running: A plant model," ProQuest Dissertations Publishing, University of Michigan, 1998.
- [10] J. Hodgins, "Planar Biped," MIT Leg Laboratory, Boston, Mass., 1985.